A Taxonomy of Browsing Methods: Approaches to the 'Lost in Concept Space' Problem

Flip Korn
Dept. of Computer Science
University of Maryland
College Park, MD
Advisor: Ben Shneiderman
In partial fulfillment of an M.S. degree

April 8, 1996

Abstract

Browsing is an interactive and exploratory process for information retrieval that has advantages over search term queries in many situations. Most browsers represent a concept space as a nested, indented tree structure or a hypertext document, but the problem is that users feel 'lost in concept space' in that they have a poor sense of where they are, where they want to go, and how to get there. Many systems attempt to rectify this by displaying the underlying hypertext directed graph, but this often results in a tangled web that is not helpful to the user. Space-filling methods, on the other hand, focus more on concepts and make the relationships (links) more implicit. Topographical cues such as size, color, and location can be employed more effectively in space-filling display methods than in structure display methods. This paper investigates a list of tasks that users should be able to accomplish within a concept space, discusses data structure representations of the concept space that are required for performing these tasks, and then evaluates and classifies a variety of existing browsing methods according to their abilities to perform the enumerated tasks.

1 Introduction

Browsing is useful and important in information retrieval. While queries are powerful methods for specific searches, they are not always helpful to the user when the information sought is vague and ill-formed or when the user can describe it only in the right context. Often the user, being unaware of the database structure, has no idea whether a search term has an appropriate scope. Browsing, on the other hand, is more exploratory and interactive and allows the user to apply recognition skills to match target concepts rather than recalling specific items.

Depending on the type of task at hand, browsing can take on a range of different functions, from specific to purposeful to serendipitous. Existing browsers vary in how focused they are, which limits their usefulness for some kinds of tasks and makes them more appropriate for other kinds of tasks. For example, The Galaxy of News, a tool discussed later on in this paper, could probably only be used for serendipitous browsing because the user does not know how to anticipate the results of his or her actions. The Galaxy of News functions more like a newspaper, delivering articles for the reader's edification rather than answering questions.

Hypertext offers the most popular method of browsing. A hypertext system is a collection of nodes, metaphorically thought of as being notecards, connected by links. Each notecard contains textual and graphical items and a number of links to other cards indicated by visual cues. Clicking on a linked item jumps to the desired notecard. Most hypertext systems require an author to design the node-link structure of a document beforehand, but there are some systems that automate this process given a linear document, the Hypertext Medical Handbook being one of them [Fri87]. Thus, methods that display a hypertext structure can, in some sense, apply to all kinds of documents.

Often the user lacks an intuitive sense of location and orientation within the hypertext structure, not knowing where s/he currently is in the document and which is the best way to go from there. This is known as the 'lost in concept space' problem. "The main disadvantages of using hypertext at present seem to be consequent of its sheer lack of physical presence and integrity. The ... flexibility of reading on screen is disorienting for a user who can't conceptualise an overview of the system." [McA87]

A solution to this problem may be to provide visual cues for the user to develop a cognitive map, or overview, of the information space. Maps allow people to navigate through a big environment by helping them abstract routes and paths they have not encountered before. In general, they make people feel less lost. Studies of how people in new, large-scale physical environments construct cognitive maps of their environment indicate that landmarks are very useful and important as markers for recognizing territories.[Dil90] Once people designate landmarks, they form routes connecting the landmarks. The global map is understood in terms of the smaller- scale route maps. While the information spaces we wish to study are non- geographic, some of the underlying cognitive theory will still apply to the metaphor we choose for representing them.

A table of contents is an example of a structure that serves the function of giving the reader an overview of a document. It hierarchically presents indices to sections and subsections, leaving out low-level details and spacing index intervals neither too close together nor too far apart. Once the user browses the table's tree hierarchy, s/he can retrieve details on demand by flipping to the page.

2 Description of Tasks

A document can be abstracted as a bunch of concepts linked together in a meaningful way. A concept space is the the view of the document presented to the user in terms of concepts, which are nodes of the underlying data structure. As previously mentioned, browsing has advantages over keyword searches for some kinds of tasks or queries. We now discuss some of these tasks and describe the data structures and browsing methods that are best suited for them. Some of these queries apply structural metrics while others employ text processing techniques on keywords in order to present a different view of the concept space to the user to aid the given task.

Knowing the general context which contains the search target, the user should be able to browse through the items and recognize what they want. For example, suppose a user with an interest in gross anatomy wants to browse concepts pertaining to the gastrointestinal system. S/he might consider consulting a National Library of Medicine publication called the Medical Subjects Heading tree (MeSH). The MeSH tree is seven levels deep and contains about 18,000 terms describing systems (digestive, cardiovascular, etc.), body regions, and tissue types. Given the context of the GI subtree, the user can browse items like the stomach, intestines, etc. There is also the reverse task of finding the context encompassing a query concept. Using a similar example, a user can gain an understanding of the how the cecum (the upper part of the large intestine) fits into the digestive system by looking at the subtree in which it is contained. These queries are most appropriate on trees because of the way the tree hierarchy aggregates nodes from general to specific, but there are analogs for graphs. Perhaps an algorithm can be used to impose a hierarchy on a graph. [RBS94]

It is sometimes useful to get a local view of a given region by seeing neighboring concepts of the query term. In a tree this includes the siblings of a given node and possibly some of its cousins and uncles. It is also sometimes useful to understand how a query term, or concept, fits into the overall global structure. Visualizing a concept node surrounded by the entire data structure representation of the concept space can show how a concept fits into the 'big picture'. For example, displaying an entire tree reveals which nodes have the most children.

Often the concepts of a concept space can be linked in a variety of ways. Each link configuration suggests a different semantic association between concepts, and thus a different view of the concept space. Browsing through neighboring nodes in a structure is not sufficient for understanding a variety of relationships between concepts. A browser should provide a way to indicate that seemingly unrelated concepts are associated in a given context. Examples of association queries are, "Which diseases have similar symptoms to X?" and "Which organs will most likely be affected when the carotid artery is torn?"

We have already mentioned that one way of browsing specific items in a specific context by looking at a node's children. Some views of a concept space don't structurally cluster related concepts. "What are the major organs of the body?' and "Summarize all major research on karposi sarcoma within the past two years," are two examples of 'survey' queries. A survey of concepts may require weaving together nodes from here and there in the structure as is the case for viewing the major organs of the body when the structure is organized by systems or body regions. The survey, or summary, can be presented spatially as a table (or map) of contents, or temporally as a guided tour.

Finally, it is often instructive to highlight nodes of a tree or graph by structural metrics, such as the number of children a parent has, the depth in the tree, or the degree of a graph vertex. This can be indicated by color, size, etc. These cues can also be used to indicate a semantically meaningful variable that is orthogonal to the context under consideration. For example, a MeSH tree displaying the digestive system might color code nodes by body region.

In summary, here are the queries, or tasks, described above:

1. know context, find specific item
2. know specific item, find context
3. neighbor; structural proximity
4. structural query: how item fits into the 'big picture'
5. associations; interrelationships
6. summary, survey, overview, 'greatest hits': filter out low priority items
7. guided tour: animated traversal of part of the structure
8. cue by a structural parameter

9. cue by a variable

3 Data Structures

Before reviewing a variety of methods for browsing a concept space, it is important to consider the data structures that these methods use for laying out the display and for performing tasks. Almost all of these methods store contents in a node-link structure ranging from (1) trees, to (2) 'tree + links', to (3) multitrees, to (4) networks.

A rooted tree is a collection of nodes and edges organized hierarchically with exactly one node at the top level (the root). Only nodes from two consecutive levels of the hierarchy are linked such that when two nodes are linked together, one node is the parent and the other is the child. This results in a defining property of trees that there is exactly one path between any two nodes. The hierarchical structure of trees relates all linked nodes by the 'ISA' relationship where sibling nodes at one level are easily abstracted as concepts aggregated under a larger concept schema at a higher level. This enables the browsing of subject headings, assuming that the user understands how his/her search targets fit into each heading.

Outliners, Harsha Kumar's tree browser, cone trees, hyperbolic trees, and treemaps are the methods for visualizing a tree that are considered in this paper. An outliner enables users to dynamically interact with a table of contents; the tree browser displays a tree as a node-link diagram with scrollbars and a smaller overview window; Cone trees display the tree in 3D;

hyperbolic display trees from a non-Euclidean space onto the plane; treemaps show hierarchical relationships by slice-and-dice 2D containment.

When a concept space is represented as a tree, top-down traversal requires the user make a decision at each 'fork in the road'. This is because each parent node is partitioned into several mutually distinct concepts, which are the node's children. A good partitioning will give the user an intuitive sense of what is 'behind each door'. At the very least, the user should be able to make an educated guess based on elimination as to which direction leads to a search target, especially at the highest levels. A good concept tree should provide the information hiding that trees are good for while choosing keywords and partitionings that give the user a cognitive map of the concept space. However, it is not always possible to compartmentalize a concept into mutually disjoint subconcepts of roughly the same size. Many subject heading trees, for instance MeSH, have keywords and even subtrees that occur more than once. A tree can reuse subtrees by using symbolic links. Thus, a concept may appear as the child of two different parent concepts simultaneously by allowing both parents to be linked to the concept. These so-called symbolic links appear the same as ordinary links when the user is browsing top-down. In this paper this data structure will be called 'tree + links' because it is tree with some extra links between nodes. IGD is a document layout system that helps the user build a tree from a directed graph where the original graph nodes become the children of the tree. Here sibling leaf nodes are linked together and internal nodes across the same level are linked by a feature called 'link inheritance' to indicate interrelationships between leaf node cousins.

Multitrees are a class of directed acyclic graphs that have the property that a tree extends both up and down from every node. Thus, the user browses a node's children to search through contents and a node's parents to search through contexts. Furnas has implemented a browser for a family genealogy multitree [Fur94].

A network is a directed graph. Links can exist between any two nodes. This provides freedom to express a greater variety of relationships between concepts than trees, but can lead to chaos because relationships can be ambiguous and edge crossings can occur. This paper describes several methods for browsing networks including Intermedia, SemNet, CYBERMAP, and "Flying Through Hypertext". Intermedia is the most basic; it does not scale up and results in a tangled web. SemNet attempts to circumvent this problem by displaying in 3D. CYBERMAP attempts to rectify the tangled web by aggregating to form a simpler network. "Flying Through Hypertext" is intended to skim the network in a guided tour.

4 Literature Review

Providing an effective overview of a concept space is a tricky task. The essential problem is that the entire concept space in its original form cannot all fit on the screen at once. The challenge is to display a meaningful abridgment of the textual and pictorial concepts of the concept space and to allow access to details by some method that does not disorient the user from the global context. Beard and Walker have empirical results to argue strongly for an overview. [BW90] The methods described below use scrolling, panning, roaming, zooming, and distortion to achieve this. Another problem is that concept spaces lack a geographical topology. Visualization requires mapping a high-dimensional space onto a two-dimensional or three-dimensional display. A common approach for hypertext documents is to draw the underlying directed graph structure. Intermedia is an example of a system that draws networks (see Figure 1). The problem with this straightforward approach is that, while it is well-suited for depicting small, sparsely connected graphs, it does not scale up. Larger and denser graphs are perceived as a tangled web. Graphs with as few as five nodes can be non-planar, which means that any layout that prevents edge crossings is impossible to draw in a plane. "For richly interconnected material or documents of a reasonable size and complexity, it is not possible to include everything in a single browser without the problem of presenting 'visual spaghetti' to the user." [Dil90]

4.1 Structure Display Methods

4.1.1 Outliners: The Oldest Trick in the Book

A text outline widget is a window pane that allows the user to dynamically interact with a table of contents. Initially only the top level of the TOC tree hierarchy is displayed. The user can expand and contract individual branches of the tree in the space of the parent node. This method is superior to the static view of the entire table of contents, as demonstrated in experiments [CS94]. Another table of contents browser that has been implemented is the multipane interface. Here local views are displayed in window panes separate from the global view pane. Users also performed better with the multipane interface.

4.1.2 Drawing the Node-Link Diagram

The "tree browser" is an implementation by Harsha Kumar of the Human-Computer Interaction Lab at the University of Maryland for manipulating trees. [Kum94] It displays a node-link representation of a tree with scroll bars for panning parts of the tree that do not fit on the screen. A miniature version of the entire tree is shown to the left of the main display acting as an overview map. The nodes of the tree are color coded based on the given task. The user can dynamically filter out nodes and links by using two-way sliders. There is a tight-coupling between actions in the main the display and in the overview map.

The Hyperbolic Browser is another tool for visualizing trees in 2D [LRP95]. The tree is laid out in the hyperbolic plane and then projected onto a circular region. By a property of the hyperbolic plane, exponentially more space is available with increasing distance from the radius of the circle. Since trees fan out exponentially, they can be laid out uniformly in hyperbolic space. The authors claim that the browser can display 1000 nodes of which about the 50 nearest the focus can show from 3 to dozens of characters of text. The figures included in their paper do not, however, verify this.

Multitrees are a subclass of directed acyclic graphs and a superclass of trees. They are different from trees in that each node may have more than one parent. One important property of multitrees is that, from each node, a tree extends both downwards towards the descendants, and upwards towards the ancestors. Looking down from any node one sees various contents; looking up, one sees various contexts. The ability to select among contexts from parent nodes has important ramifications for browsing. For one thing, the same node can be viewed differently depending on its context. In fact, Furnas has implemented a multitree browser that is somewhat analogous to Kumar's tree browser [Fur94]. His browser looks like a tied knot with the single node in focus at the center of the screen with the ancestor tree extending to the left and the descendant tree to the right. This is what he calls the "centrifugal view" It turns out that multitrees have nice visual properties. While they are more general than trees, all layouts can avoid edge crossings. Also, the union of the trees for two adjacent nodes is a topological tree and thus planar. This allows for a smooth animation sequence to indicate a new node coming into focus. Multitrees are particularly well suited for when fragments of a tree hierarchy are reused in other hierarchies. However, fragments composed in one context may not work well in another context.

Cone trees [RMC91] (see Figure 3) offer a way to view a tree structure as a three-dimensional vertical (or horizontal) tree from top to bottom with uniform spacing between levels. Child nodes hang off their parent in a cone formation that resembles a Christmas tree. Due to the three-dimensionality of the display, some nodes may be occluded from view, but can be accessed by rotating the cone tree in real-time.

While controlable and reversible animation in three-space is very engaging, cone trees waste a lot of screen space. Display allocation is lopsided – due to the geometry of the cone, the bottom of the screen is overutilized, (especially where the leaf nodes are) and the top of the screen is underutilized. Cone trees do a poor job of representing content information. The tree nodes are too small to contain much text. The current implementation does not use visual information cues such as color and size. But these changes probably would not contribute much because of the waste of screen space.

SemNet [FF88] displays the directed graph structure of an information space in three dimensions and gives the user the ability to translate, rotate and zoom in three-space with a mouse in real-time. Sizes of nodes are scaled according to their distances from the camera. Several spatial positioning methods were compared: random, isolation of three 'natural' orthogonal variables to be used as axes, placement of nodes by number of connectivities, and WYSIWYG-edited layouts. Unfortunately, SemNet only creates a tangled web in 3D. The panning and zooming capabilities are engaging but they only seem to make the display more confusing and intractable.

In observing that one can gain insight into the organization, size, depth, and detail of a book just by flipping through it, Manber and Lai implemented a prototype to allow users to fly through a hypertext document [ML91]. The hope is that a brief guided tour will orient the user within a hypertext framework. This is consistent with the kind of browsing one might do in a library. People like to pull books off the bookshelf to rifle through because the title the table of contents do not always give one an idea of the scope of the book. Their program traverses a hypertext digraph using a variety of graph algorithms, such as bread-first search and depth-first search, depending on the browsing function. They represent the map of the hypertext as a maze and animate the traversal in the maze. Unfortunately, the class of graph representations of a maze doesn't include any non-planar graphs or graphs of degree greater than four so this representation doesn't generalize. Besides, the maze representation does not seem to have much more of an advantage over graph representations.

4.1.3 Aggregation

While there are algorithms for converting hypertext digraphs into trees [BSR92], the order of the links may be semantically different from what the author intended. We consider methods that scale down the original structure without modifying the link order.

Because "subjects need what can be called 'terrain' knowledge in addition to 'street' knowledge," [McA87] Ray McAleese displays both a regular-sized version of the graph in a window with horizontal and vertical scroll bars and, alongside it, a scaled-down version of the original graph for an overview. McAleese calls the graphs that represent knowledge spaces "concept maps" because the nodes contain phrases that describe concepts and are linked to other nodes by meaningful relationships. He attempts to rectify the tangled web problem by compressing some of the nodes and links into more general nodes, called "filing cabinets" That way complex meshes of concepts can be aggregated into a single schema. Unfortunately, he does not describe a general algorithm for performing this on directed graphs. One is left to guess that he is merely expanding and contracting branches of a tree because his example uses a tree hierarchy. For directed graphs, particularly in hypertext structures, there are methods for creating aggregates of nodes by looking at biconnected and strongly connected components of the graph [BS91].

Peter Gloor expands on the idea of consolidating nodes and links with some interesting ideas but a rather disappointing implementation in his CYBERMAP tool [Glo94] (see Figure 2). CYBERMAP ignores the original node-link structure of a hypertext document and instead generates new nodes and links based on indexing techniques that represent articles of the document as keyword vectors. It first partitions the information space into clusters based on the relative similarities of articles. These clusters, which are called "hyperdrawers" (similar to McAleese's "filing cabinets" become nodes of what will be a network display of the document. Hyperdrawers obey the simple properties that there is an even distribution of information among the hyperdrawers and that the the user is not overloaded by the total number of hyperdrawers. CYBERMAP tries to spatially position the hyperdrawers on the screen so that related nodes are close together by starting at the center of the screen and placing unrelated nodes towards the edges. It utilizes the fisheye principle by highlighting the currently-selected node and suggesting possible successor nodes by placing nodes with a high degree of interest adjacent to the current node. Another important aspect of the system is that, in addition to employing indexing techniques to generate a generic overview, a model of the user is taken into account to give the layout a more personalized view. At the beginning of a session the user enters a list of keywords of particular interest to him or her. Also, the user's interactions with the system are monitored to dynamically link and unlink nodes.

In a few examples CYBERMAP performed well, but it seems to do poorly the larger and the less ordered the underlying document is. The network display is not very effective when the number of nodes is large because too much information gets absorbed in each hyperdrawer.

Interactive Graphical Documents (IGD) [Fei88] is a system that deals with the tangled web problem by assisting the user in converting an existing hypertext structure into a tree where the pages of the original document are the leaf nodes and internal nodes, or chapters, are created to subsume the pages, and metachapters are created to subsume chapters in a hierarchical fashion. It then uses a principle called "link inheritance" to consolidate links with common ancestors into one single link, reducing the overall number of connections and simplifying the display. Only links between pages of the same chapter and single links to indicate if any links exist between pages of different chapters are shown. Thus, links that have little importance in a local context are suppressed. The created hierarchy is displayed as a tree of chapters eventually fanning out to the leaf node pages of the document. Subchapters, which are initially hidden, can be made visible in nested set notation by clicking on the respective chapter within which it is spatially contained. Traversal up or down the hierarchy can be accomplished by zooming in on a subchapter to replace the screen display space in which it was contained. This is IGD's main drawback. Another drawback is that it does not support the conversion of an existing document into a tree structure – an author must use a display direct manipulation editor to write a new tree-structured document. There are, however, algorithms that can automate this process with some success [RBS94].

4.2 Space-Filling Display Methods

Because "a pixel is a terrible thing to waste," the following methods attempt to pack detailed overview maps of a document space onto one screen. This is important for allowing access to details on demand without losing the global context.

4.2.1 Painting the World Map

A treemap is a space-filling approach to visualizing hierarchical information utilizing 100 percent of the screen space (see Figure 4). Treemaps were developed by Ben Shneiderman of the Human-Computer Interaction Lab at The University of Maryland. In addition to presenting structural information, they can display content both textually and visually with cues such as color, box size, and location. Given a hierarchy such as a file system, tree maps represent the structure in nested-set notation by spatially containing child nodes inside their parents. The screen is partitioned into rectangular boxes with children recursively contained within their parents possibly displaying the entire tree. Each level of the tree is rotated 90 degrees from the previous level in a slice-and-dice fashion. It turns out that the slightly different problem of using Euler circles (Venn Diagrams), ovals or even rectangles that are not sliced-and-diced is an NP-Complete problem because optimizing screen space requires bin packing.

The main strength of treemaps is that they provide access to detail while preserving the global context. No scrolling, panning, or zooming is needed. "The number of nodes that can be displayed by a treemap can be an order of magnitude greater than traditional graphical tree diagrams." [TJ92] But treemaps have some limitations. For trees with many levels, displaying the entire contents at once can overload and disorient the user. Lower- levels become too small to reveal any content information. While determining bounding box size by a weight metric provides a nice cue, it also pushes smaller nodes off the screen to make room for the larger ones. In general, the greater the disparity between bounding box aspect ratios, the worse treemaps perform.

Xia Lin uses Kohonen's feature map algorithm to organize the information space for a space-filling overview layout [Lin93] (see Figure 5). Here the entire screen is utilized to display what he calls a semantic map of labeled concept areas of size proportional to the density of the keyword concept. These concept areas are bounded orthogonal polygons that contain the keyword and any co-occurring keywords with frequence above a given threshold. Kohonen's algorithm is used to parse a document into an undirected planar graph where the nodes are frequently used keywords and a link represents a strong co-occurrences between keywords. This planar graph is displayed in its dual form as a geographic-like map. This graphical layout is intended to preserve semantic relationships between related items and reveal the topology of the information space. Lin's semantic maps fare at least as well as user-authored maps and far better than randomly-generated layouts in retrieval tasks.

4.2.2 Distorting the View

A fisheye lens is a very wide angle lens that shows places nearby in detail while also showing remote regions in successively less detail. A fisheye view display simulates this effect. "The fundamental motivation of a fisheye strategy is to provide a balance of local detail and global context." [Fur86] Sarkar and Brown experimented with graphical fisheye views of graphs and maps

[SB92] (see Figure 6). Instead of zooming into regions of a graph or map that are too large to fit on one screen, the user can view local regions in detail under a magnifying glass. The area allocated for each node or region is proportional to its degree of interest, which is a function of its a priori importance and its distance from the current focus (where the magnifying glass is currently positioned). Fisheye views can be beneficial when applied to some domains. But for graphical views, the distorted geometry produced by fisheye transforms looks unnatural and can be very disorienting.

The Galaxy of News [Ren94] is a tool for browsing a database of newspaper articles. It initially displays a global overview of the information space as keywords from the articles that were considered the most general and important by some parsing algorithm. Each keyword is an umbrella for hidden keywords that refer to more specific topics. Also displayed are blue branch-like links to the child keywords from each parent keywords. The child keywords of a parent become visible when the user zooms in on that parent. This process recursively repeats until it bottoms out at the actual articles, which are the leaf nodes in the tree. Thus, at each level the user is zooming in on a topic of interest while simultaneously filtering out the other topics (zoom-and-replace) to eventually obtain the desired details. This layout is known as pyramidal structuring and is one of the strengths of the system.

The Galaxy of News system is divided into four layers. The bottom two are the newsbase of articles with human-authored annotations, and the relationships between articles and topics, which can either be authored by a user or can be generated by a parsing algorithm. These layers constitute the back-end of the system. The front-end of the system deals with the spatial construction of the information space and the visual cues for user interaction. The actual location placement of keywords in the global overview is computed via a recursive constraint-based grid structure algorithm. The derivation of the keyword pyramid is accomplished via an Associative Relation Network algorithm.

The Galaxy of News has very smooth panning and zooming capabilities. The allocation of keywords in the pyramid very nicely clusters the information space for displaying high-level overviews, although the fish-eye layout does not give the user a sense of the densities of the information topics. The global overview and the navigation in space metaphor make a large information space with no natural topography seem more tractable, and the fading out of parent keywords helps preserve the global context. The Galaxy of News also seems to be a nice tool for discovering subtle relationships between articles by bringing two keywords into focus at the same time. However, the main limitation of the system is that it is only useful for serendipitous browsing. It is hard to envision the system being used to perform specific goal-oriented tasks. This makes it unsuitable for information retrieval from a digital library.

5 Evaluation of Methods

Below is a list of each of the methods described in the previous section. The list included my subjective judgement of what tasks, or query types, each is most appropriate for performing:

1. outliner: summary; know context, find specific; structural query

2. tree browser: browse neighbors, cue/filter by content variable

3. cone tree: structural query

4. hyperbolic tree: structural query

5. multitree: know specific, find context; know context, find specific; neighbors

6. filing cabinets: neighbors, summary

7. CYBERMAP: neighbors, summary

8. IGD: interrelationships; know context, find specific

9. treemap: summary; filter by structural parameter; filter by variable; structural query

10. semantic map: summary; interrelationships

11. fisheye view: know context, find specific; filter by structure; structure

12. Galaxy of News: summary; interrelationships; know specific, find context

13. flying through hypertext: guided tour

Structural queries are for understanding how a node or region fits into the 'big picture'. The methods that try to pack the tree on the screen (treemaps, hyperbolic trees, cone trees) and fisheye views are appropriate for this task because they attempt to show the entire structure on the screen at once, allowing the user to view the region of intereset surrounded by the global context.

Survey queries ask for a summary of the concept space. Both semantic maps and Galaxy of News present frequently occuring keywords on a 2D display. An outliner can present a summary overview the way a table of contents does. Filing cabinets and CYBERMAP aggregate nodes and links into fewer nodes and links to derive a smaller overview map.

Semantic maps, Galaxy of News, and IGD are all tools that indicate interrelated concepts. Semantic maps do this in the positioning of its polygon boundaries and placement of co-occuring keywords within these boundaries. The Galaxy of News can be used for this purpose when two keywords are brought into focus simultaneously. IGD shows interrelationships via 'link inheritance', which are single links between chapters, and links between pages within the same chapter.

Because of the way trees are structured, they are good particularly for finding a specific concept within a given context, and can be employed for the reverse of finding the context from the concept. Multitrees are particularly well-suited for finding the context from a given concept, and The Galaxy of News is capable of this, too.

Of all the methods reviewed, only "Flying Through Hypertext" is suited for giving guided tours.

6 Discussion

An astronomical increase in the amount of information being put on-line today poses challenges for information retrieval. The information superhighway paves the way to making dreams like the 'library of the future' possible. Information visualization techniques will play an important role in making this data amenable to users.

In this paper we first investigated methods of displaying a hypertext structure. Considering the connectivity of any large network such as the

World Wide Web, this approach will not suffice although refinements and improvements may help. I believe the way to go is to construct hierarchical, space-filling overview maps of the concept space that may ignore the 'true' structure of the document. Even something as simple as a hierarchical index has had immediate success on the Web. Consider the impact that resources such as Yahoo (**http://www.yahoo.com**) and The Whole Internet Catalog (**http://nearnet.gnn.com/wic/newrescat.toc.html**) have had in helping users 'surf' the internet. Trees are excellent structures for storing and indexing exponentially increasing information. A tree with 30 children for each parent can allow access to any one node in almost 1 million nodes in less than 4 mouse clicks and to almost 1 billion nodes in less than 6. But it is not always possible to compartmentalize a concept space into a rigid hierarchy while preserving its semantics. Sometimes nodes that are closely interrelated wind up in separate subtrees, but are linked only by a distant ancestor.

We have investigated several space-filling and other methods for displaying a document space. They utilize the entire screen space and present visual cues to take advantage of the awesome human perceptual abilities. This, combined with allowing users to dynamically interact with the display, allows them to zoom in on some regions and filter out some regions for access to desired information.

References

[BS91] R. Botafogo and B. Shneiderman. Identifying aggregates in hypertext structures. In *Proceedings of Hypertext '91*, pages 63–74, New York, December 1991. ACM.

[BSR92] R. Botafogo, B. Shneiderman, and E. Rivlin. Structural analysis of hypertext: Identifying hierarchies and useful metrics. *ACM Transactions on Information Systems*, 10(2):142–180, 1992.

[BW90] D. Beard and J. Walker. Navigational techniques to improve the display of large two-dimensional spaces. *Behaviour & Information Technology*, 9(6):451–466, November 1990.

[CS94] R. Chimera and B. Shneiderman. An exploratory evaluation of three interfaces for browsing large hierarchical tables of contents. *ACM Transactions on Information Systems*, 12(4):383–406, October 1994.

[Dil90] A. Dillon. Navigation in hypertext: A critical review of the concept. In *Proceedings of IFIP Interact '90*, North-Holland, 1990.

[Fei88] S. Feiner. Seeing the forest for the trees: Hierarchical display of hypertext structure. In *Conference on Office Information Systems*, pages 205–212, New York, NY, 1988. ACM.

[FF88] K. Fairchild and G Furnas. Semnet: Three-dimensional graphic representations for large knowledge bases. In Raymonde Guindon, editor, *Cognitive Science and its Applications for Human-Computer Interaction*, pages 201–233. Lawrence Erlbaum, Hillsdale, NJ, 1988.

[Fri87] M. Frisse. Searching for information in a hypertext medical handbook. In *Proceedings of Hypertext '87*, pages 57–66, New York, NY, November 1987. ACM.

[Fur86] G. Furnas. Generalized fisheye views. In *Proceedings of CHI '86*, pages 16–23, New York, NY, 1986. ACM.

[Fur94] G. Furnas. Multitrees: Enriching and reusing hierarchical structure. In *Proceedings of CHI '94*, New York, NY, April 1994. ACM.

[Glo94] P. Gloor. *Exploring Large Information Spaces*. PhD thesis, University of Zurich, Switzerland, 1994.

[JS91] B. Johnson and B. Shneiderman. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Proceedings of IEEE Visualization '91*, pages 284–291. Kaufman & Nielson Ed., October 1991.

[Kum94] H. Kumar. Visual information management for network configuration. Technical report, University of Maryland, Department of Computer Science, College Park, MD, June 1994.

[Lin93] X. Lin. *Self-organizing Semantic Maps as Graphical Interfaces for Information Retrieval*. PhD thesis, University of Maryland, College of Library and Information Sciences, College Park, MD, 1993.

[LRP95] J. Lamping, R. Rao, and P. Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of CHI '95*, New York, NY, May 1995. ACM.

[Mar87] Gary Marchionini. An invitation to browse: Designing full-text systems for novice users. *The Canadian Journal of Information Science*, 12(3):69–79, 1987.

[Mar92] Gary Marchionini. Interfaces for end-user information seeking. *Journal of the American Society for Information Science*, 43(2):156–163, 1992.

[McA87] R. McAleese. The graphical representation of knowledge as an interface to knowledged based systems. In *Proceedings of IFIP Interact '87*, North-Holland, 1987.

[McA89] R. McAleese. Navigation and browsing in hypertext. In Ray McAleese, editor, *Hypertext: theory into practice*, pages 6–44. Ablex, Norwood, NJ, 1989.

[ML91] U. Manber and P. Lai. Flying through hypertext. In *Proceedings of Hypertext '91*, pages 123–132, December 1991.

[RBS94] E. Rivlin, R. Botafogo, and B. Shneiderman. Navigating in hyperspasce: Designing a structure-based toolbox. *Communications of the ACM*, 37(2):87–96, 1994.

[Ren94] E. Rennison. Galaxy of news: An approach to visualizing and understanding expansive news landscapes. In *Proceedings of UIST '94*, pages 3–12, Marina Del Rey, CA, 1994.

[RMC91] G. Robertson, J. Mackinlay, and S. Card. Cone trees: Animated 3d visualizations of hierarchical information. In *Proceedings of ACM SIGCHI '91*, pages 189–194, April 1991.

[SB92] M. Sarkar and M. Brown. Graphical fisheye views of graphs. In *Proceedings of CHI '92*, pages 83–91, April 1992.

[TJ92] D. Turo and B. Johnson. Improving the visualization of hierarchies with treemaps: Design issues and experimentation. In *Proceedings of IEEE Visualization '91*, pages 124–131. Kaufman & Nielson Ed., October 1992.